Probabilistic Intelligent Agent Approach to Design of Alerting Systems

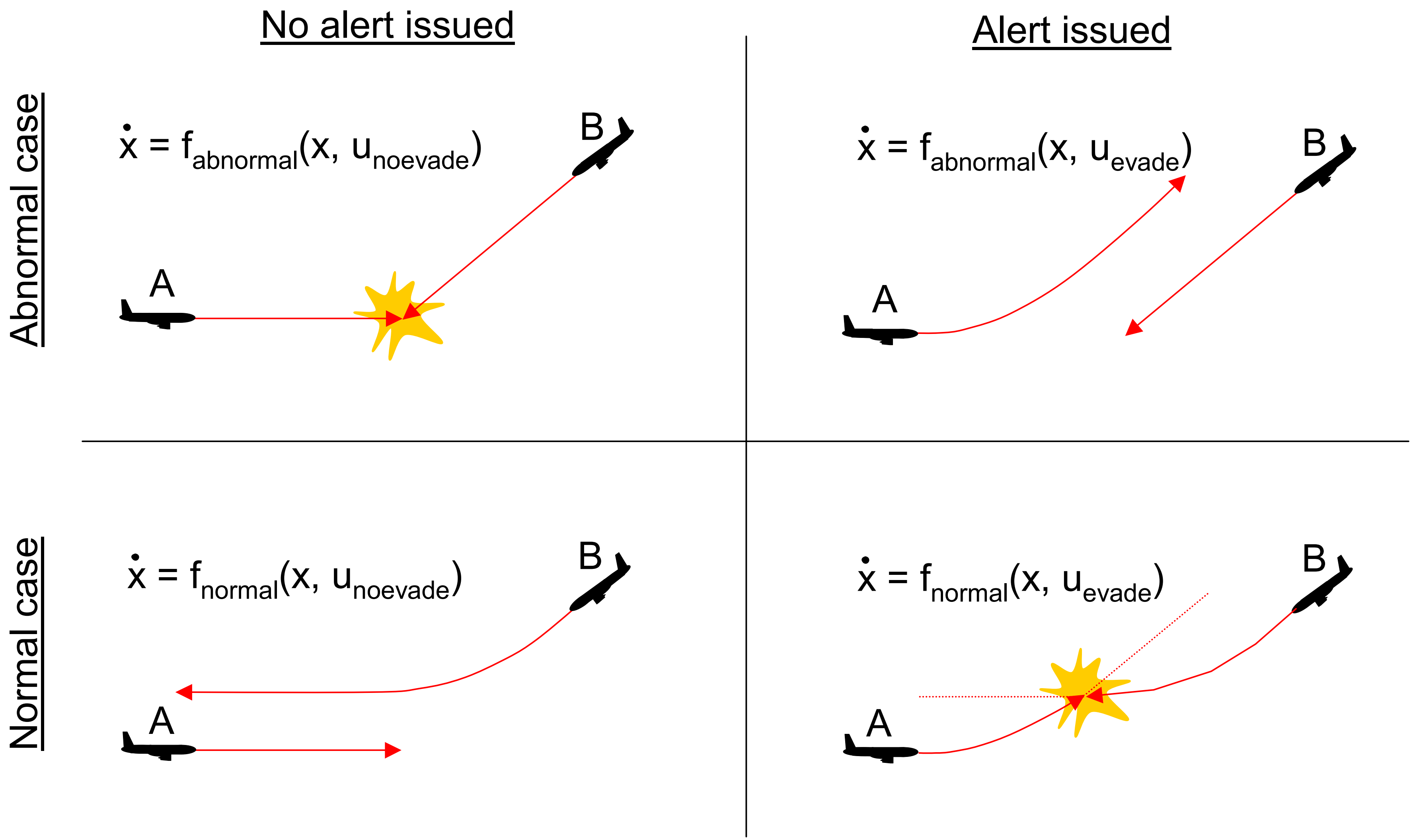
Lee Winder
Jim Kuchar
International Center for Air Transportation

FAA/NASA Joint University Program
Ohio University
June 13, 2002

Alerting Systems

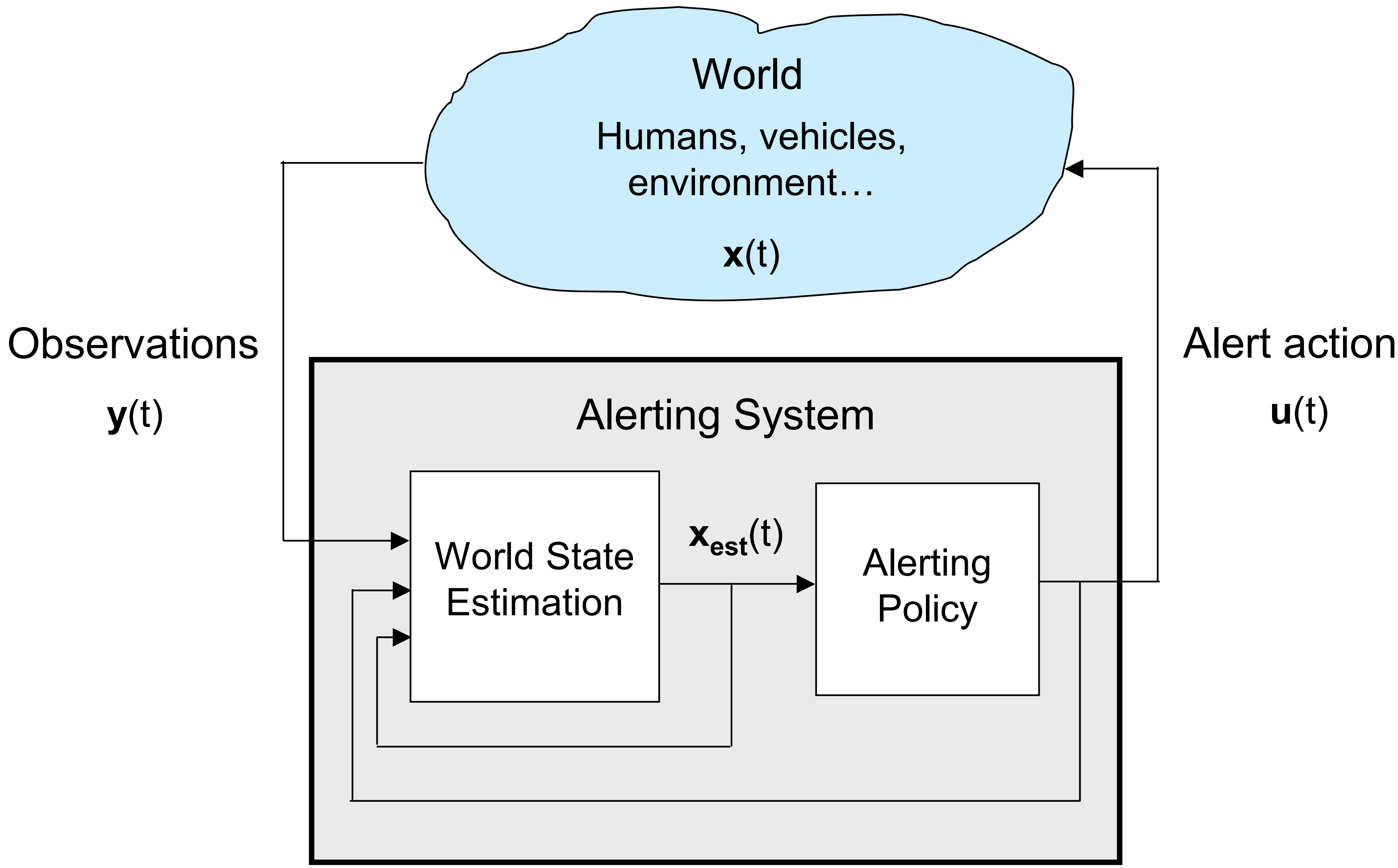
Alerting System: Automation that monitors another system and issues alert guidance to human operators when necessary to avoid some category of incident.

Importance of World Dynamics to an Alerting Decision



A good world dynamic model is critical in choosing the alert action

Generalized Alerting System as Intelligent Agent*



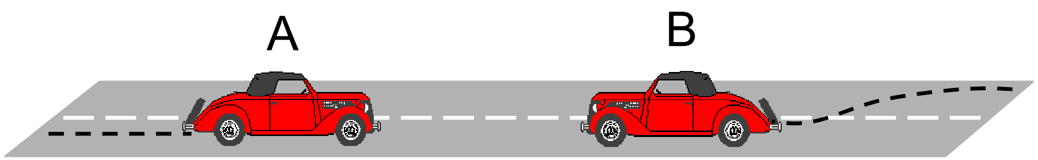
Two phases of alerting: world state update and alert action selection

(*Diagram adapted from Kaelbling, et. al.)

World Model Class Parameter: Car Encounters

Class p_1 : Failure of lane-keeping by A or B

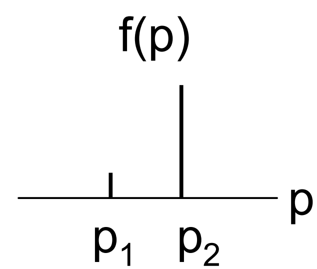
$$\dot{\mathbf{z}} = f_{fail}(d, \dot{d})$$



Class p_2 : Normal lane-keeping by A and B

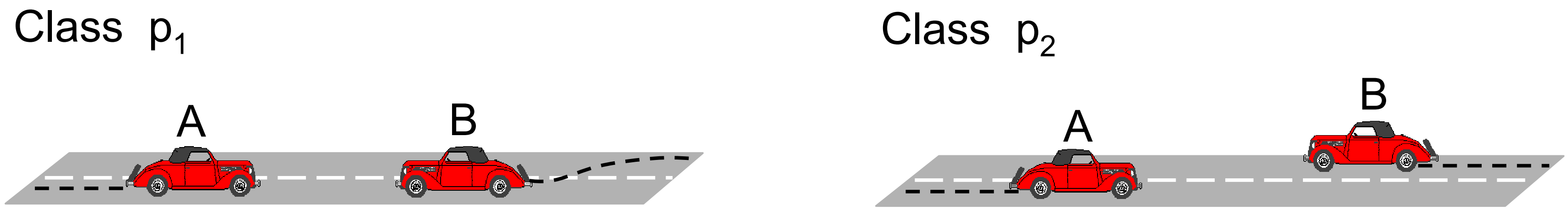
$$\dot{\mathbf{z}} = f_{norm}(d, \dot{d})$$

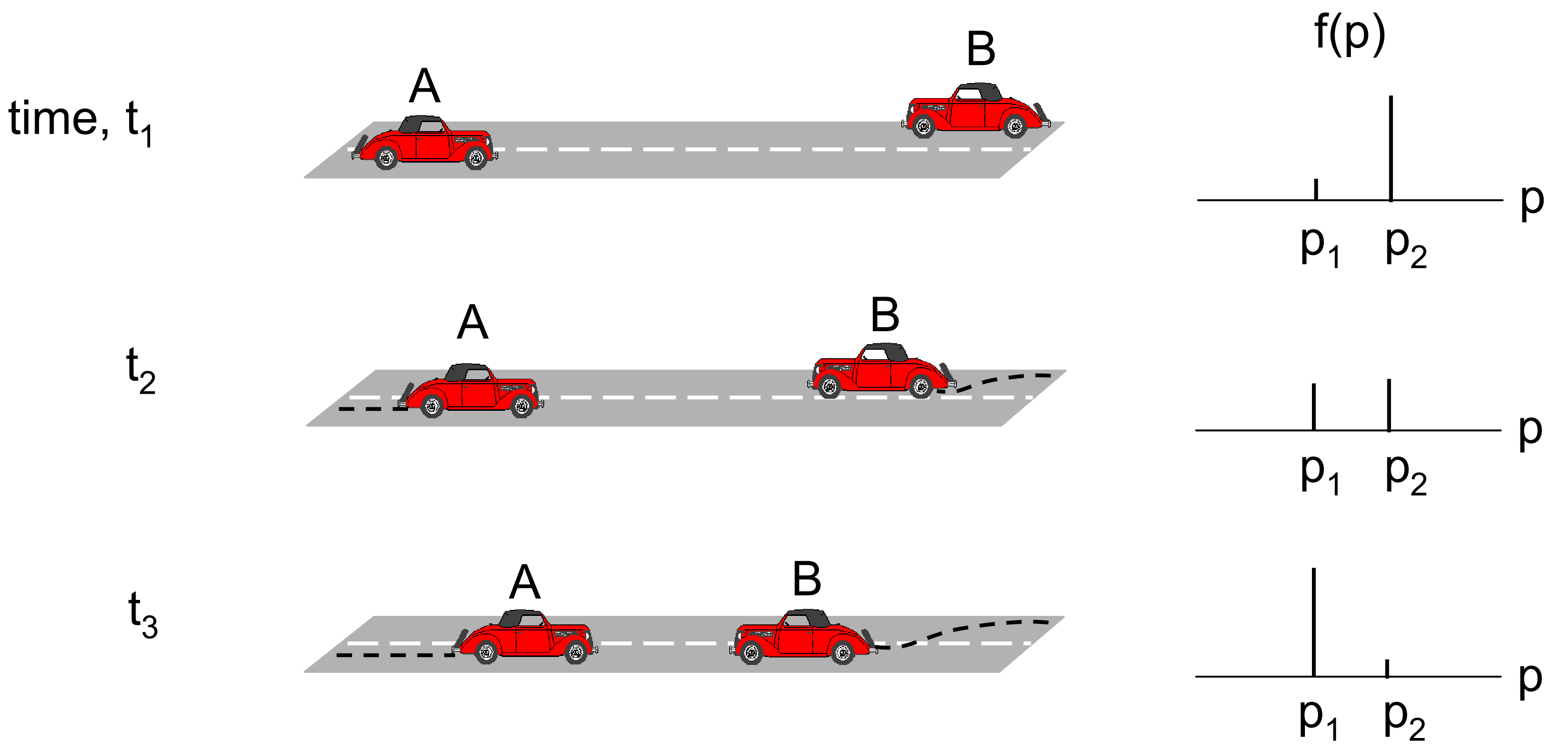
$$\dot{\mathbf{z}} = f(d, \dot{d}, p) = \begin{cases} f_{fail}(d, \dot{d}), \text{ if } p = p_1 \\ f_{norm}(d, \dot{d}), \text{ if } p = p_2 \end{cases}$$



$$x_{est} = \{ f(d, \dot{d}, p) \}$$

Class Parameter Distribution Updating





Bayes Updating of Class Parameter Distribution

Assuming Markov system → Require knowledge of previous state only
→ Update recursively

$$f_k(p_i) = \frac{\overbrace{f(z_k \mid z_{k-1}, u_{k-1}, p_i)}^{\text{Class-conditional transition likelihood}} \; \overbrace{f_{k-1}(p_i)}^{\text{Current } i^{th} \text{ class probability}}}{\sum_{j=1}^{n} f(z_k \mid z_{k-1}, u_{k-1}, p_j)\, f_{k-1}(p_j)}$$

Probability of p_i at time step k

- Recursive solution also exists if the observation is an uncertain y, from $f_{obs}(y \mid z)$, rather than z directly

Testbed System with Distinct Dynamic Classes

- Defined simple testbed system and alerting logic to
  - Implement and demonstrate Bayesian updating of class parameter distribution
  - Link choice of alert actions to class parameter distribution

- Testbed: Planar 2 “vehicle” encounter scenario

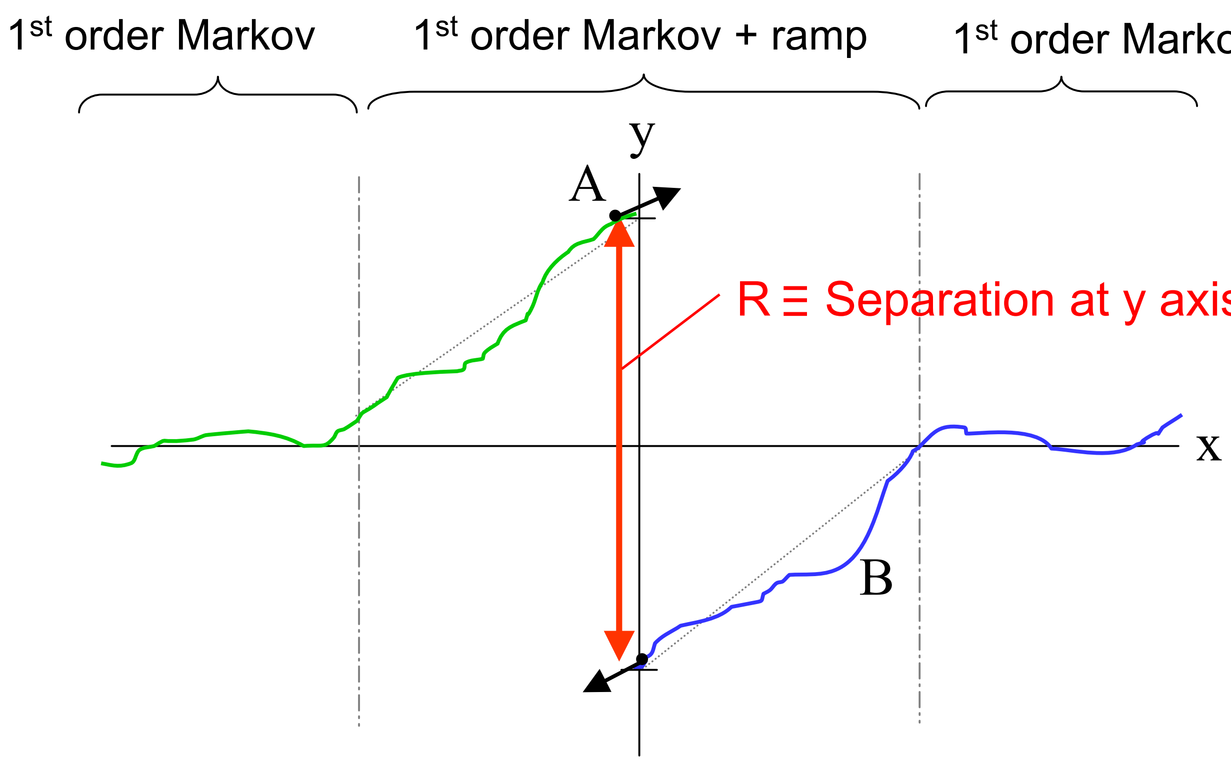
- Class parameter: 3 future trajectory classes
  - Normal, $f_o(p_{normal}) = .9$
  - Vehicle A failed, $f_o(p_{Afail}) = .05$
  - Vehicle B failed, $f_o(p_{Bfail}) = .05$

$f_o(p)$

- $f(z_k \mid z_{k-1}, u_{k-1}, p_i)$ is from a defined Markov process

Testbed System with Classes

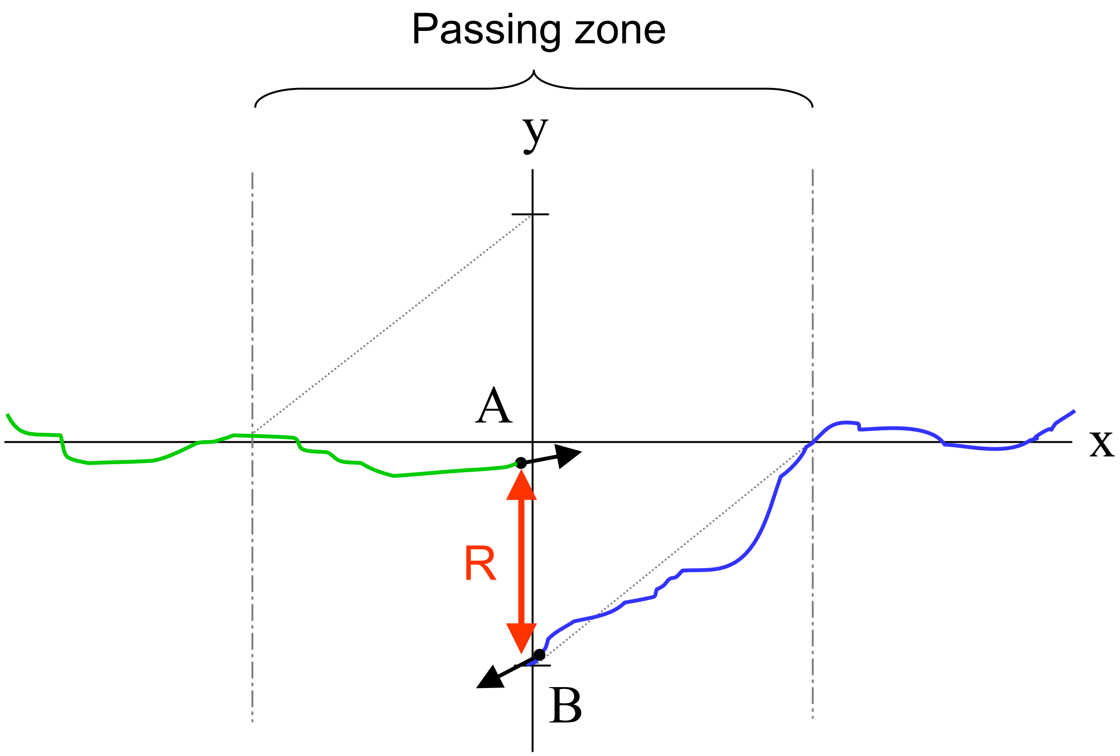
Normal Class: Vehicles approach from sides, and mean paths ramp apart so they tend to pass safely. Vehicles are responsive to maneuver commands.



(Collision if $R < R_{collision}$)

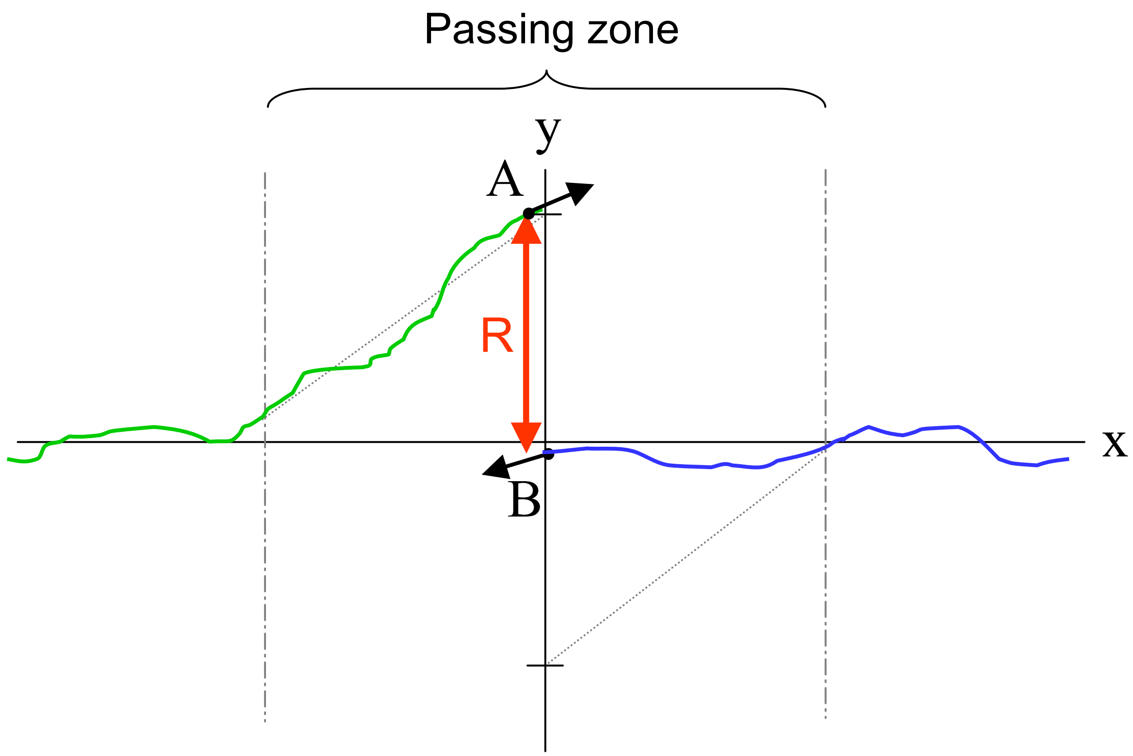
Testbed System with Classes

<u>A Failure Class</u>: Vehicle A disregards passing procedure and maneuver commands (if issued)

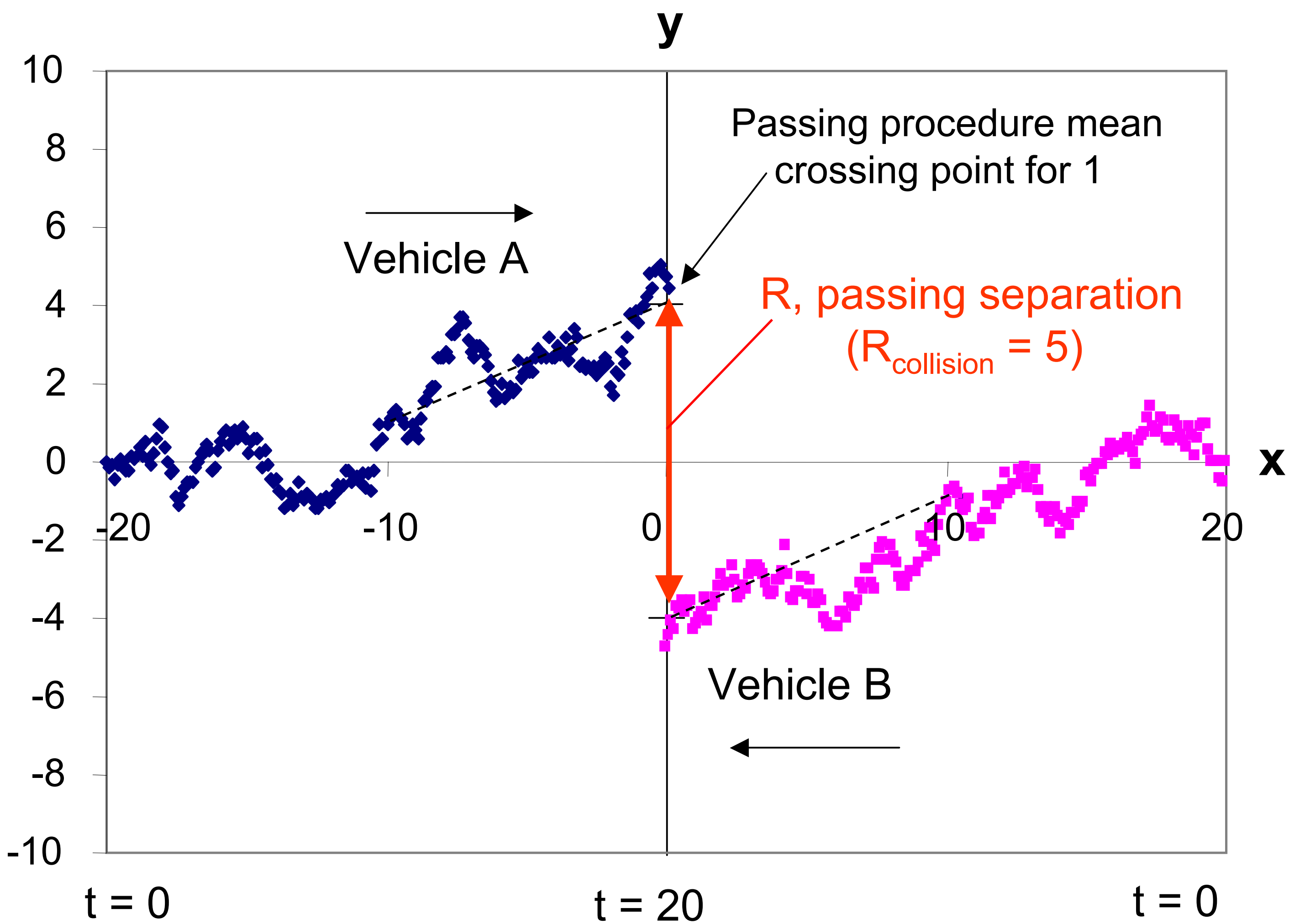


Testbed System with Classes

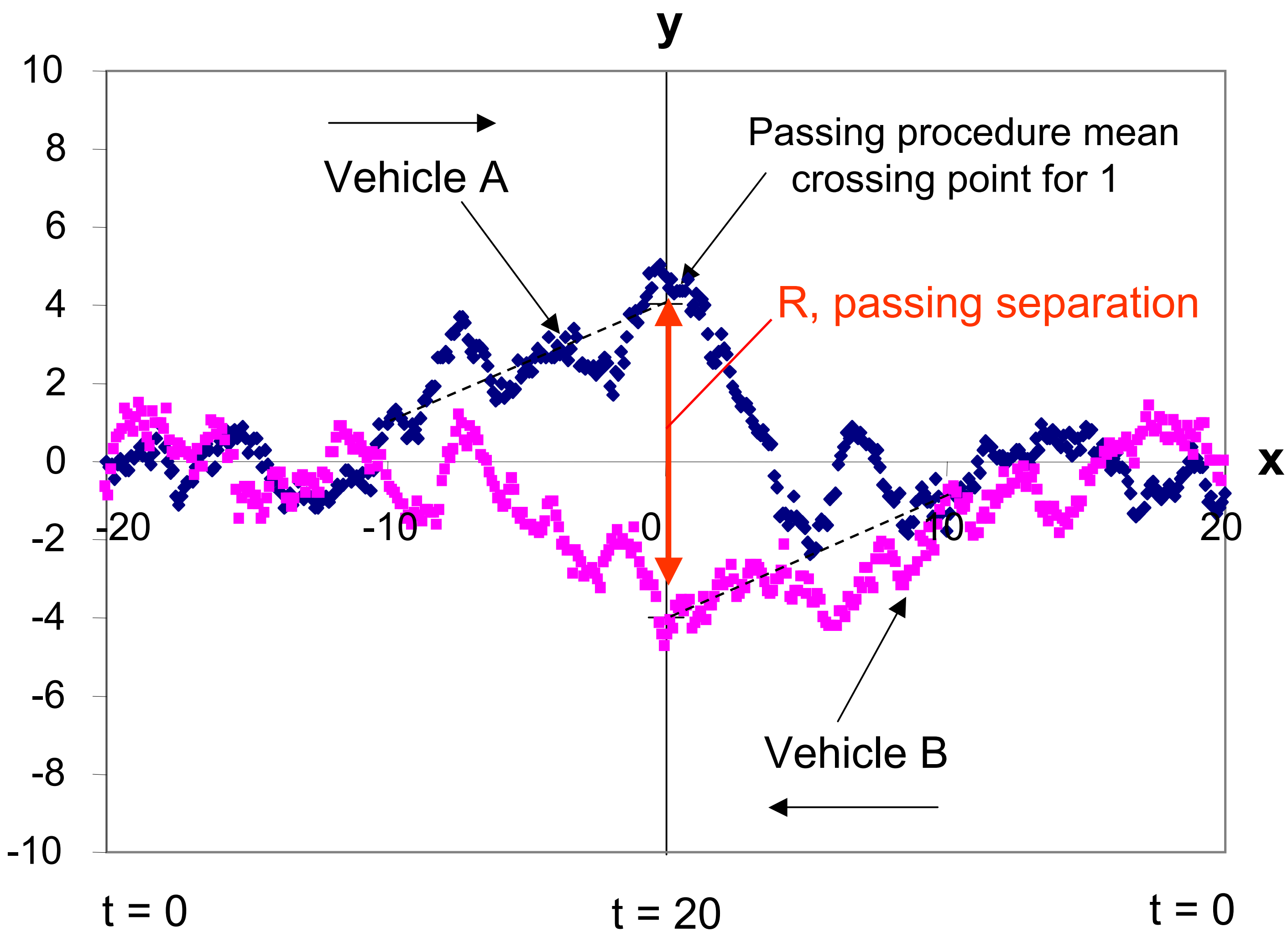
<u>B Failure Class</u>: Vehicle B disregards passing procedure and maneuver commands (if issued)



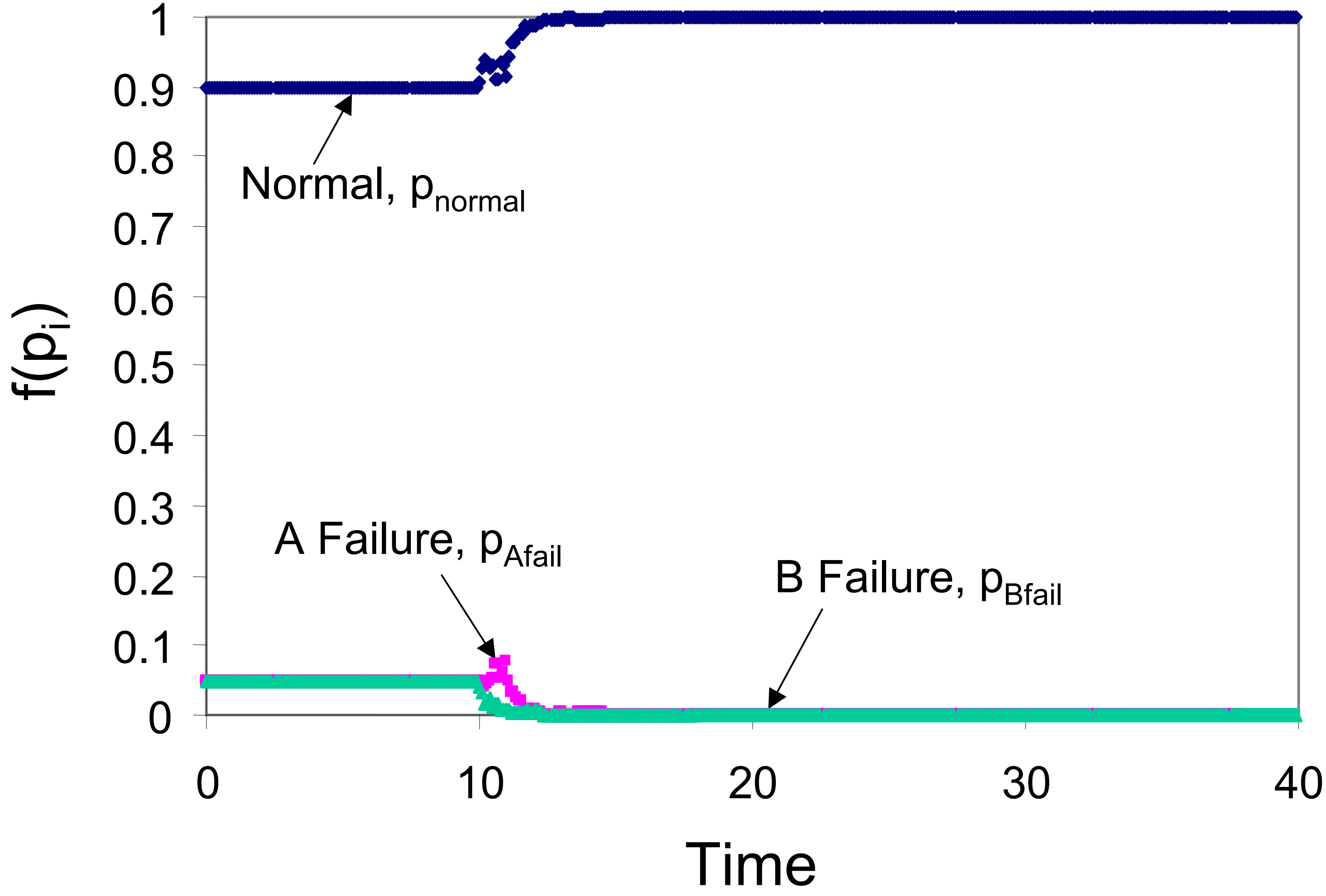
Simulated System Trajectory (Class p_{normal})



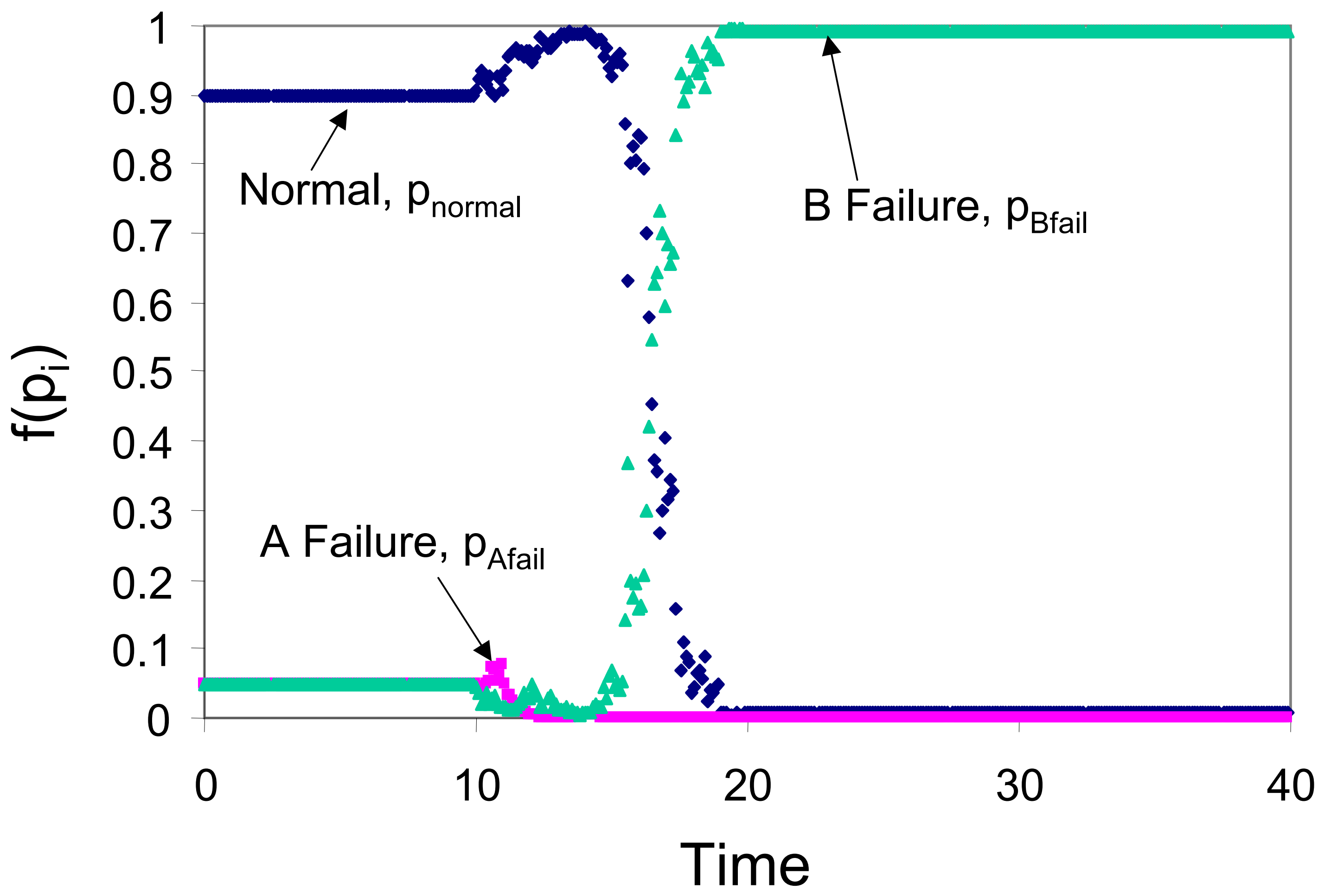
Simulated System Trajectory (Class p_{normal})



Bayes-Updated Class Distribution Trace (Class p_{normal})

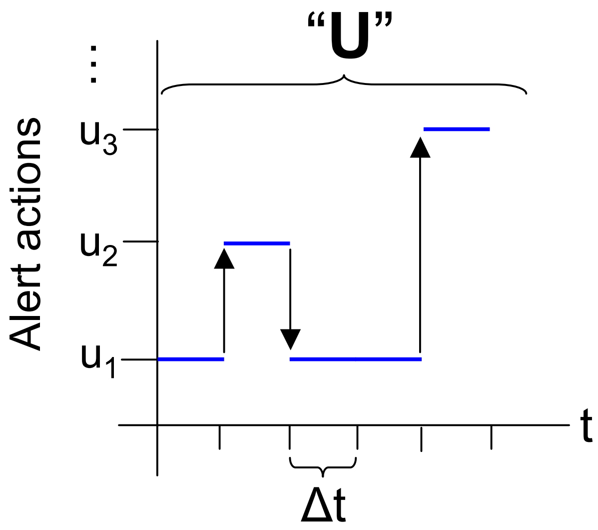


Bayes-Updated Class Distribution Trace (Class p_{Bfail})



Discrete Action Sequence, U

- U is a sequence of future alert actions.



$$U = \{ u(1), u(2), u(3), \ldots \}$$

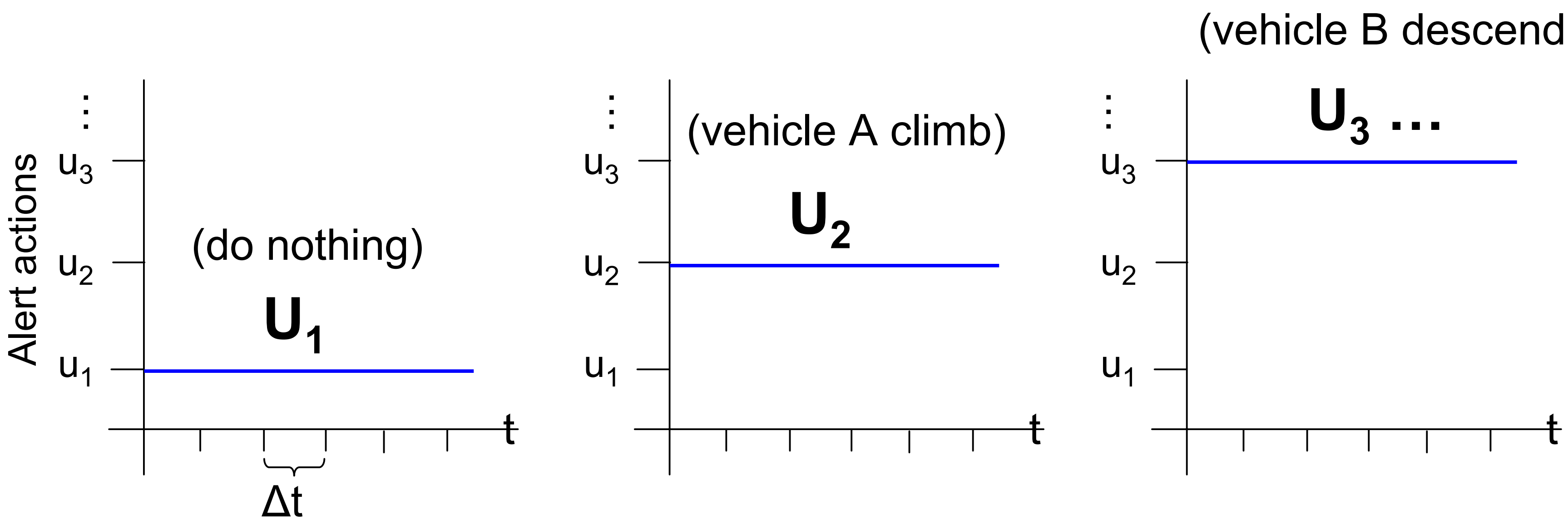
$$u(\bullet) \in \{ u_1, u_2, \ldots u_n \}$$

(e.g. A climb, B descend, do nothing…)

Limiting Scope of Action Sequence, U

- Difficult to consider all sequence options
  - For testbed system, considered a finite set of control sequences: Those of constant “u”.

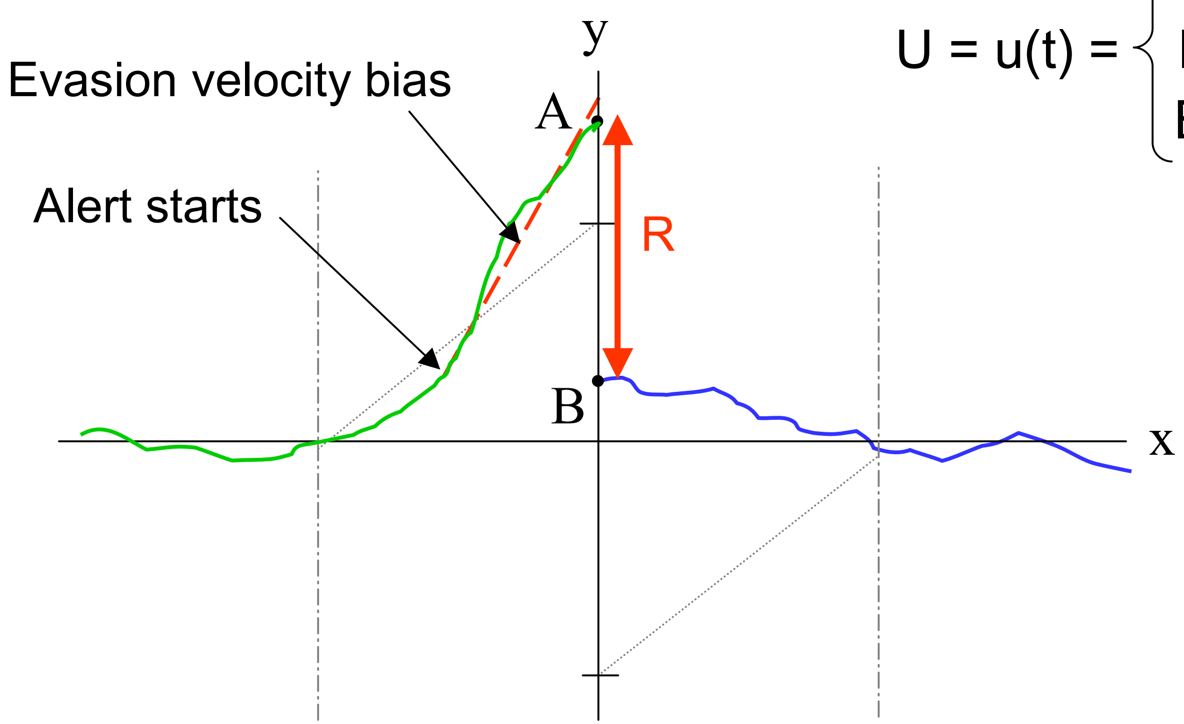
- Reduced U set:



Testbed System Action Sequence (Evasion Maneuver) Set

Evasion maneuver: Apply a constant vertical velocity bias in place of nominal behavior.

$$U = u(t) = \begin{Bmatrix} \text{No alert} \\ \text{Bias on A} \\ \text{Bias on B} \end{Bmatrix}$$



Policy derived under assumption that future u(t) = constant

Calculating Expected Utilities for Testbed Policy

- For testbed system, defined
  - Utility of any collision = 0
  - Utilities of having no collision with each alert option, U_i:

  $v_{noalert}$, v_{Abias}, v_{Bbias} and said $v_{noalert} > v_{Abias}$ and $v_{Abias} = v_{Bbias}$

- Expected utilities for each U:

$$E(\text{Utility of } U_1) = v_{noalert}\,[P(\text{no collision} \mid U_1)] = v_{noalert}\,[1 - P(\text{collision} \mid U_1, x)]$$

$$E(\text{Utility of } U_2) = v_{Abias}\,[P(\text{no collision} \mid U_2)] = v_{Abias}\,[1 - P(\text{collision} \mid U_2, x)]$$

$$E(\text{Utility of } U_3) = v_{Bbias}\,[P(\text{no collision} \mid U_3)] = v_{Bbias}\,[1 - P(\text{collision} \mid U_3, x)]$$

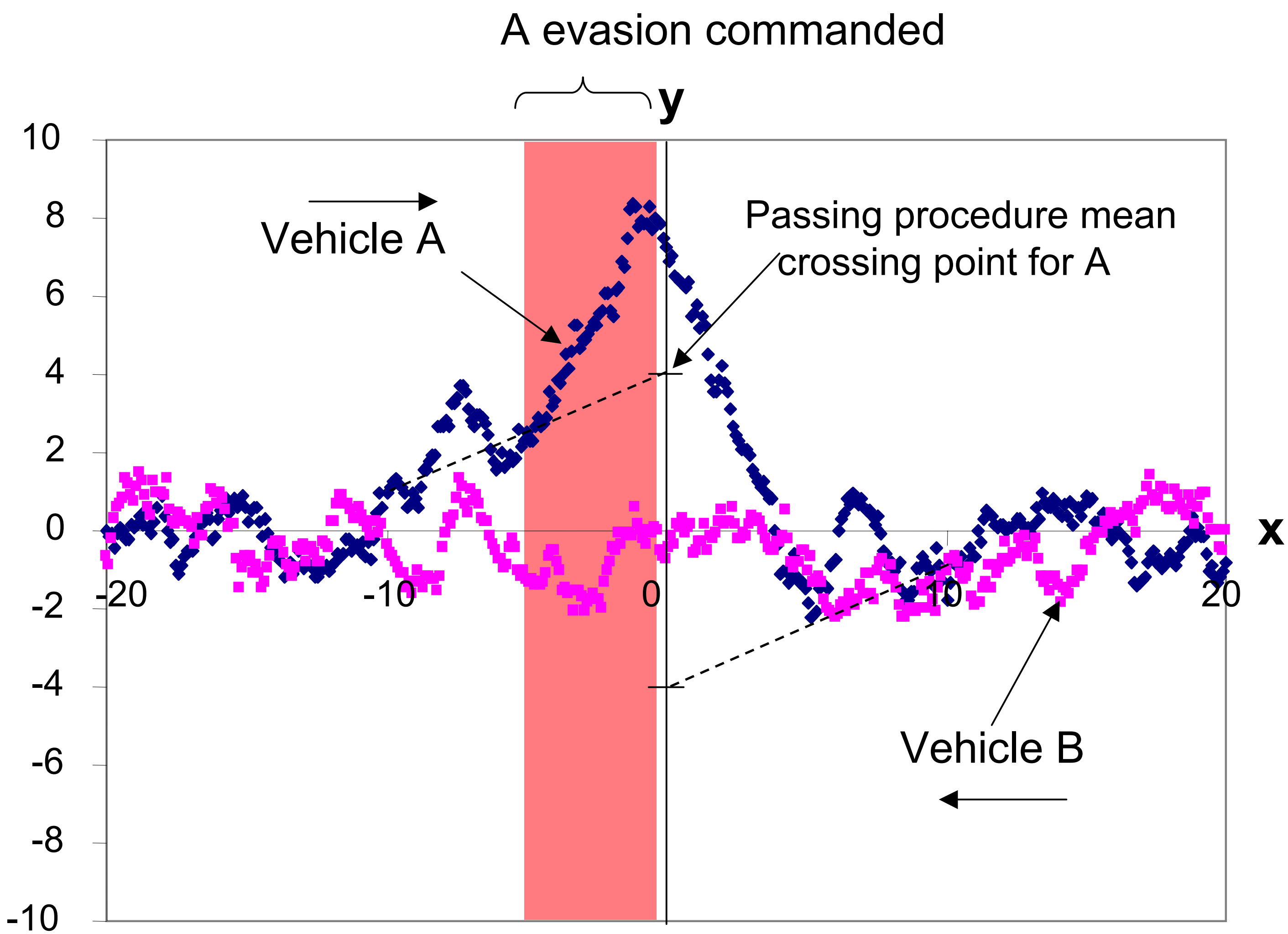
- Logic: Choose U_i to maximize expected value

Collision Probabilities from Parameter Distribution

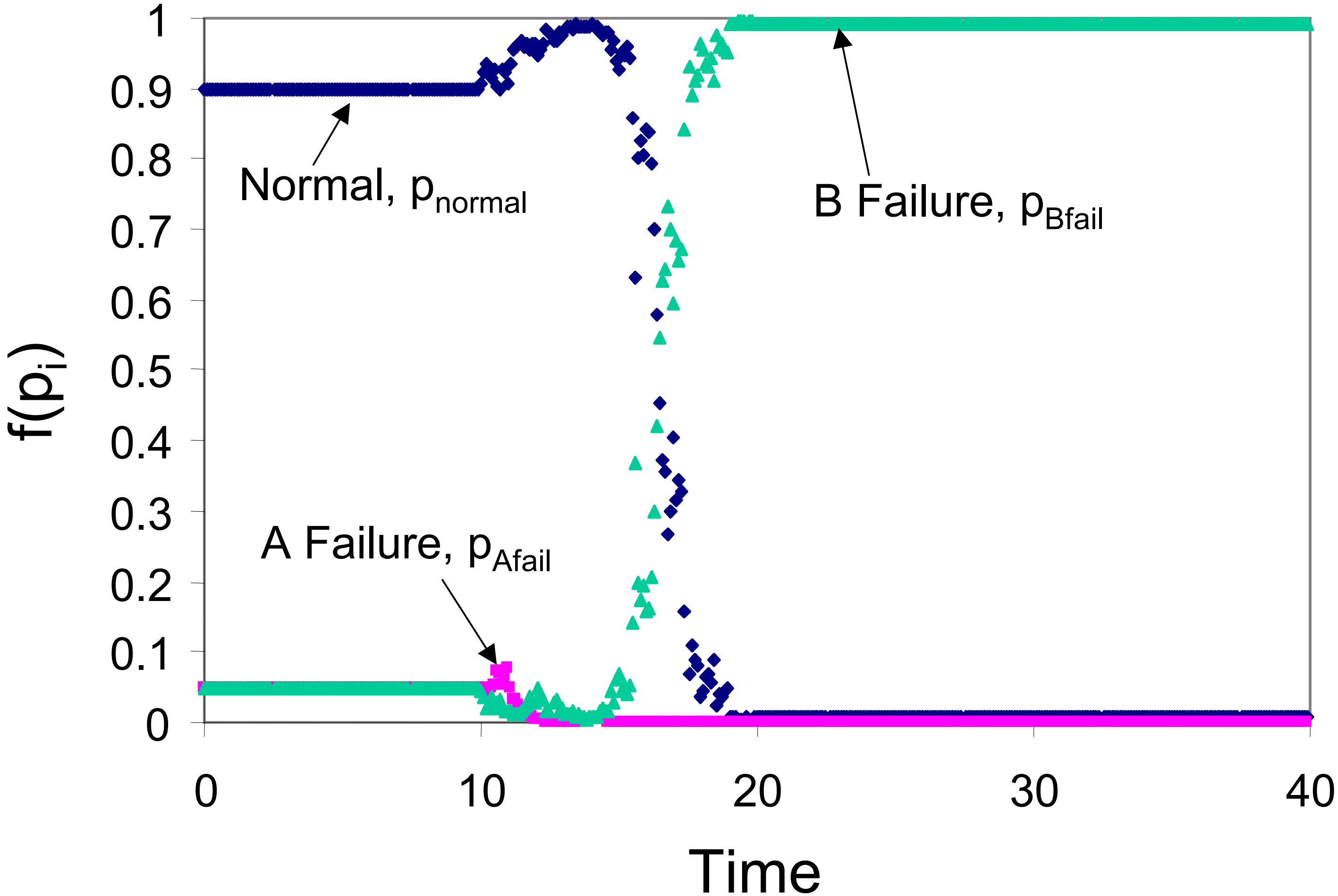
Compute probability of a collision (C) for each U

$$\begin{bmatrix} P(\text{collision} \mid U_1, x_{est}) \\ P(\text{collision} \mid U_2, x_{est}) \\ P(\text{collision} \mid U_3, x_{est}) \end{bmatrix} = \underbrace{\begin{bmatrix} P(C \mid U_1, p_{normal}, z) & P(C \mid U_1, p_{Afail}, z) & P(C \mid U_1, p_{Bfail}, z) \\ P(C \mid U_2, p_{normal}, z) & P(C \mid U_2, p_{Afail}, z) & P(C \mid U_2, p_{Bfail}, z) \\ P(C \mid U_3, p_{normal}, z) & P(C \mid U_3, p_{Afail}, z) & P(C \mid U_3, p_{Bfail}, z) \end{bmatrix}}_{\text{Class-conditional collision probabilities}} \underbrace{\begin{bmatrix} f(p_{normal}) \\ f(p_{Afail}) \\ f(p_{Bfail}) \end{bmatrix}}_{\text{Class distribution}}$$

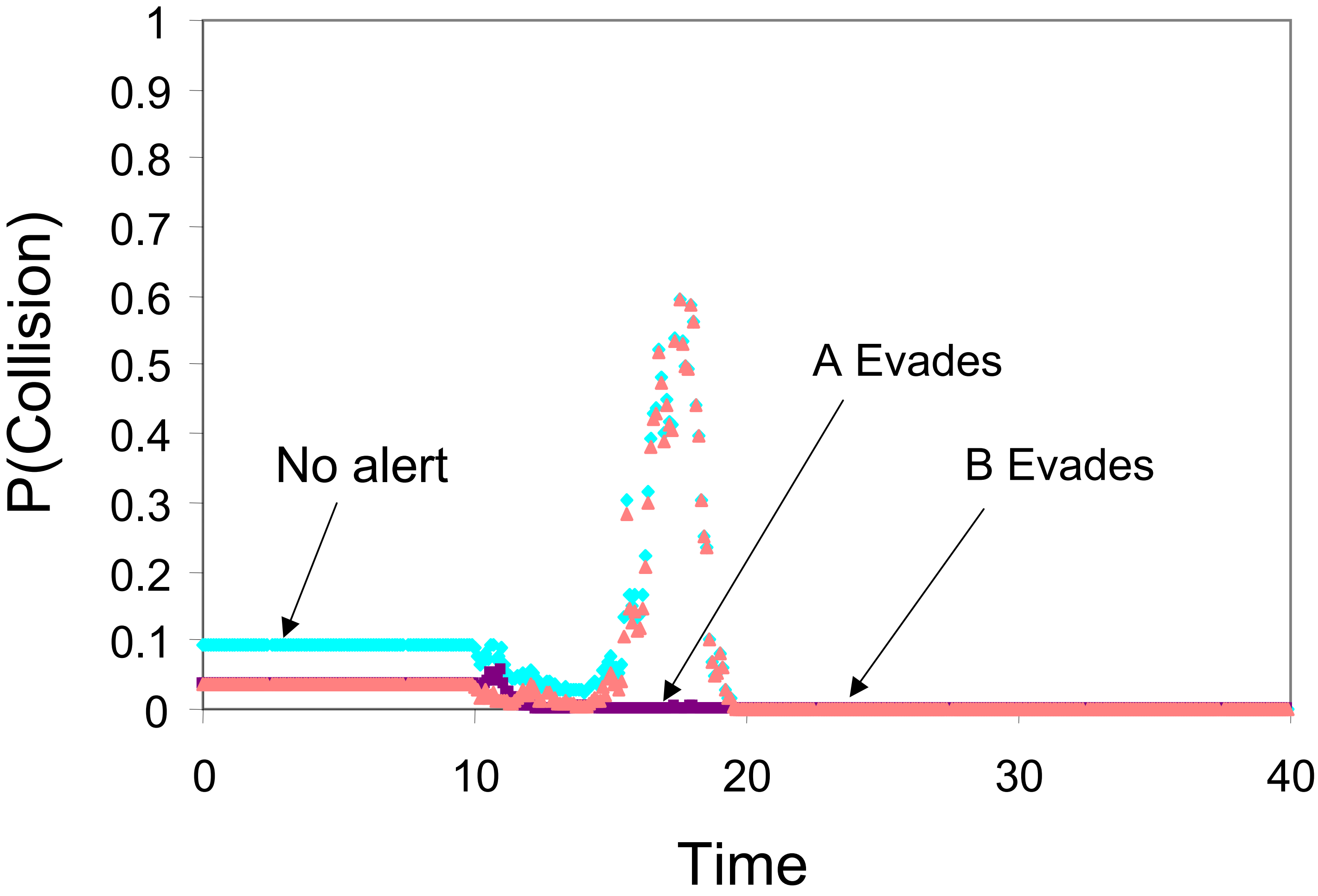
Simulated System Trajectory (Class p_{Bfail})



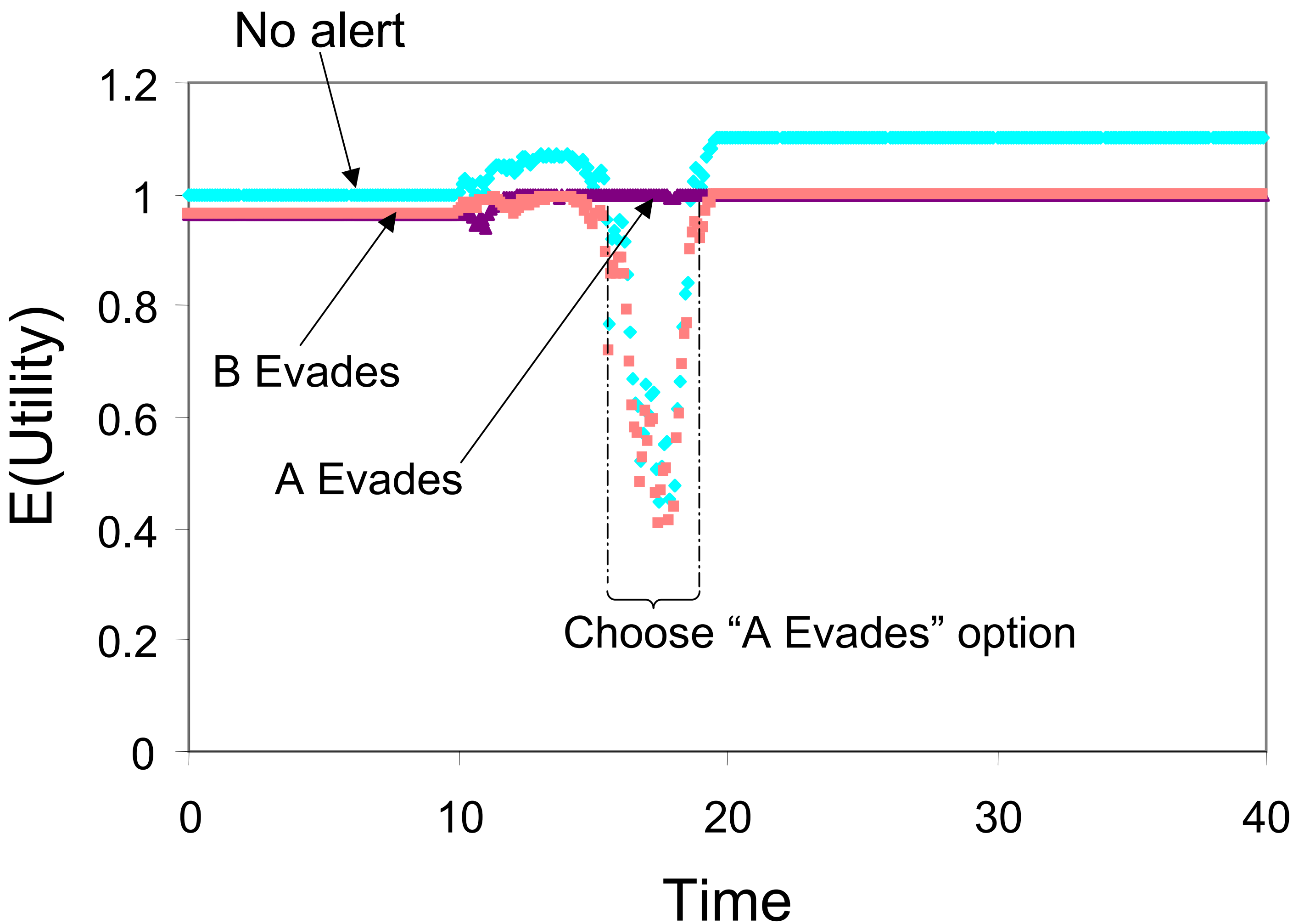
Bayes-Updated Class Distribution Trace (Class p_{Bfail})



Collision Probability Trace for Each U



Expected Utilities for Each U



$$E(\text{Utility of } U_i) = v_{U_i}\,[1 - P(\text{collision} \mid U_i)]$$

Summary to Date

- Utility and probability-based framework allowing for dynamic classes and class distribution updating

- Testbed system with similarities to some difficult alerting problems
  - Multiple vehicles/humans requiring coordination
  - Identifiable classes of trajectories
  - Dynamic resolution guidance desired

- Initial alerting logic for testbed system

Future Work

- Apply distribution update techniques to more complex and realistic alerting problems

  E.g.

  – Car collision avoidance (intersections, lane incursions, run-off-road)

  – Aircraft collision avoidance (En route, parallel approaches, runway incursions)

- Consider more sophisticated alerting policies

  – More complete sets of alert sequences, U

  – Policies that consider the value of anticipated information

    - Dynamic Decision Networks

- Compare model with existing alerting systems and concepts, identify differences and benefits, refine model

The End

Automatic Alerting Systems

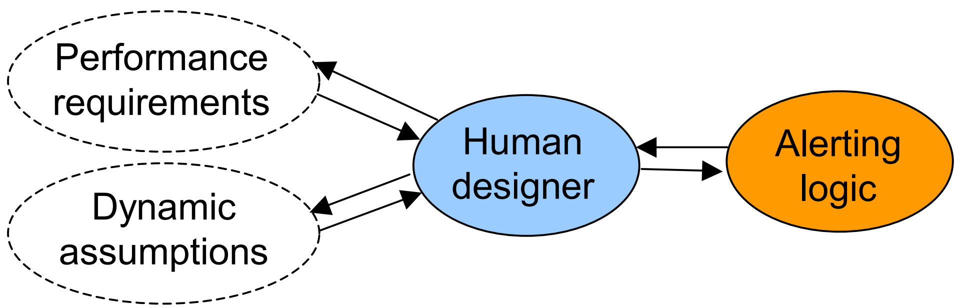
- Existing alerting systems (in aviation) target
  - Mid-air collisions
  - Controlled flight into terrain
  - Collisions due to parallel approach blunders
  - Wind shear accidents
  - Many other hazards

- Trend toward more alerting systems and more complex algorithms
  - Alert to avoid future hazard
  - Availability of computing power and state information
  - Desire to increase or maintain safety levels

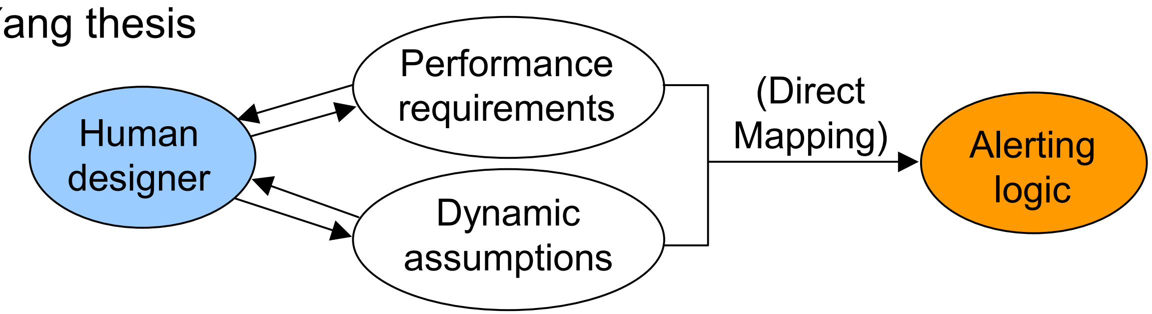
Design of Alerting Logics

- Most logics evolve from simple forms
  - Start with a simple baseline alert-triggering logic
  - Change incrementally so system behaves as desired in test scenarios
  - Field the system and adjust later to minimize user complaints

  e.g. TCAS: 10 years from concept to fielding
  10 years of adjustment in field (to version 7.0)



- Would rather logic follow directly from explicit assumptions and requirements
  - Reduce design costs, bring logic closer to "optimal"?
  - Yang thesis



Objectives

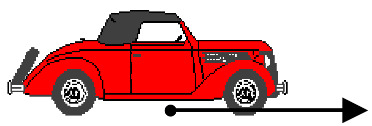
- Develop a novel design methodology applicable over a range of difficult alerting problems
  - Multiple humans/alerting subsystems to be coordinated
  - Dynamic resolution guidance desired
  - Identifiable distinct dynamic classes in the observed system (e.g. normal and failure)
    - Incorporate knowledge of structure in the nominal system
      - Procedures, rules of the road…

- Show agreement with and any advantages over evolved solutions to given problems (which presumably exhibit approximate "correct" alerting behavior)
  - TCAS
  - Proposed parallel approach alerting logics
    - Independent approaches
    - Dependent approaches
  - Others…

World State (x) Meaning

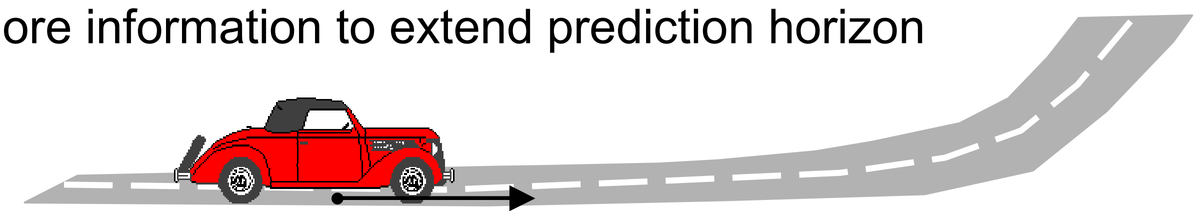
<u>World State</u>: Information about the world sufficient to know the future state or state distribution. (Markov state)

Short term prediction



x = { position, velocity }

Need more information to extend prediction horizon



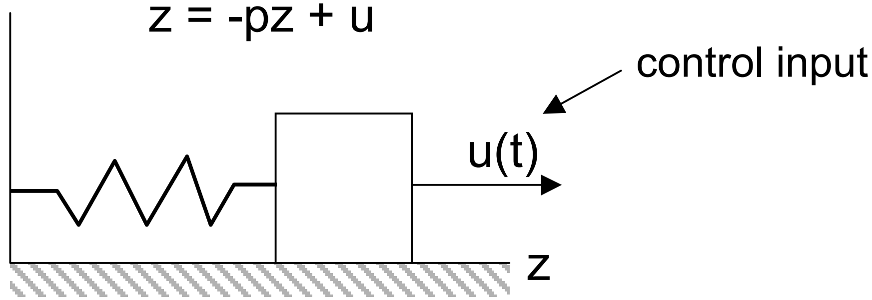
x = { position, velocity, road parameters, driver status… }

Some state variables may not be directly observable (e.g. driver status)

World State Estimate (x_{est}) is a Probability Density Function over x

Illustration: Spring-mass system with uncertain parameter

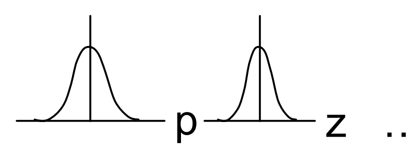
$\ddot{z} = -pz + u$



- $x = \{ z, \dot{z}, p \} \rightarrow x_{est} = \{ \hat{z}, \hat{\dot{z}}, \hat{p} \}$? No, $x_{est} = \{ f(z, \dot{z}, p) \}$

$f_0(z, \dot{z}, p)$

- Assume we know the prior distributions at t = 0:



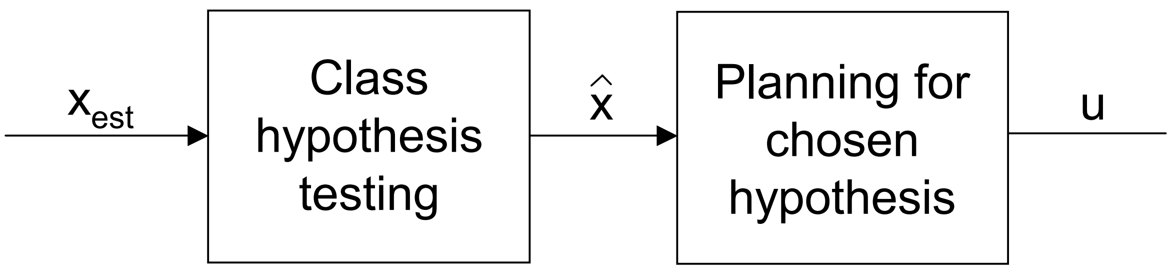
- Est. for for t>0: $x_{est}(t) = \{ f_t(z, \dot{z}, p) \}$ where $f_t(z, \dot{z}, p)$ is the posterior distribution of at time t

$= \{ \tau, f_0(z, \dot{z}, p) \}$ where τ is trajectory $\{ z(t), u(t) \}$ for all $t<0$

Alerting Policy Design

- At each time step, choose the best action for the current world state estimate

- Options



Chosen method:

Policy for Testbed System: Simple Utility-based Approach

- Employ Maximum Expected Utility principle on a set of possible action sequences, $\{ U_1, U_2, U_2 \ldots \}$
  - Update world state estimate
  - Determine expected utility of different U's from current state
  - Choose the action with the highest expected utility

$$\max_i \Big[\underbrace{\sum_j P(\text{outcome } j \mid U_i)\, \text{Utility}(\text{outcome } j)}_{E[\text{ Utility of } U_i\,]} \Big]$$

- Resembles resolution strategy of some logics (TCAS)

- Doesn't account for value of anticipated observations

General State Distribution Updating for Markovian System

For world state x, observation y, control input u

$$\hat{f}(x_k) = \sum_{X} P(x_k \mid x_{k-1}, u_{k-1})\, f(x_{k-1})$$

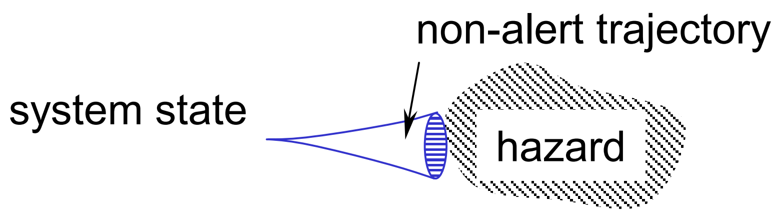
$$f(x_k) = \alpha\, P(y_k \mid x_k)\, \hat{f}(x_{k-1})$$

- $f(x_k)$ is the updated state distribution
- α is a normalization constant

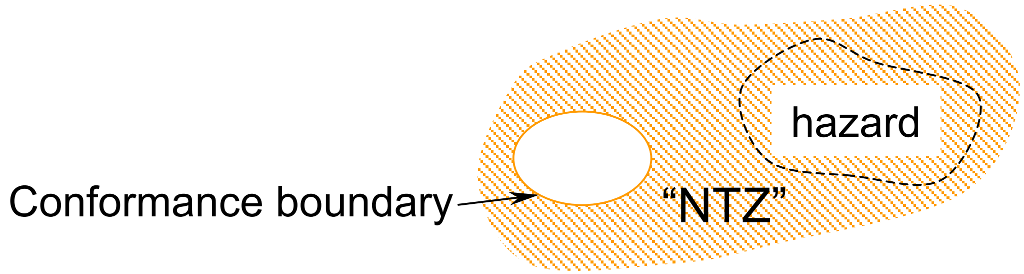
(From Russell & Norvig, AI, A Modern Approach)

Typical “Baseline Logics”

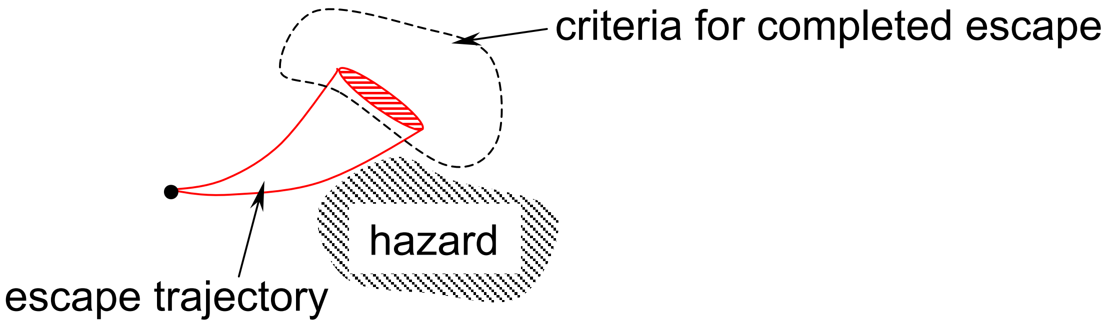
- Alert to avert a hazard (AILS, TCAS?)



- Alert when system fails to conform (PRM)



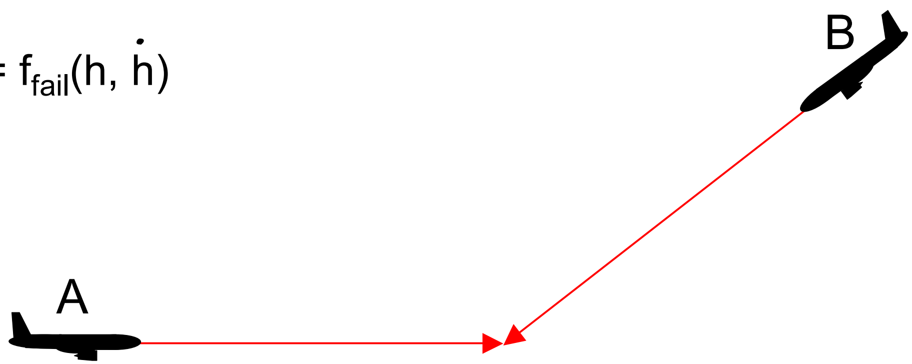
- Alert before evasion options are lost (Carpenter, Tomlin)



Discretely Distributed World State Parameter: Aircraft Encounter Classes

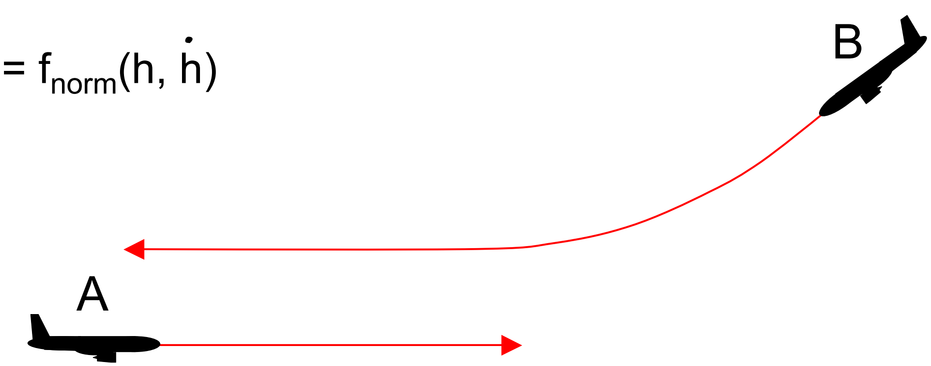
Class p_1 : A and B lose vertical separation

$\dot{\mathbf{z}} = f_{fail}(h, \dot{h})$

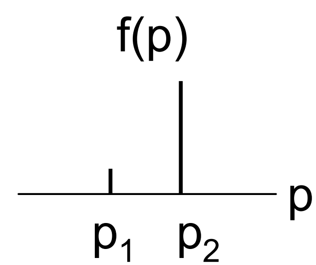


Class p_2 : A and B maintain vertical separation

$\dot{\mathbf{z}} = f_{norm}(h, \dot{h})$



$$\dot{\mathbf{z}} = f(h, \dot{h}, p) = \begin{cases} f_{fail}(h, \dot{h}), \text{ if } p = p_1 \\ f_{norm}(h, \dot{h}), \text{ if } p = p_2 \end{cases}$$



$x = \{ f(h, \dot{h}, p) \}$

State Parameter Distribution Updating